

0 About this document

Gödel's famous proof [2, 1] is highly interesting, but may be hard to understand. Some of this difficulty is due to the fact that the notation used by Gödel has been largely replaced by other notation. Some of this difficulty is due to the fact that while Gödel's formulations are concise, they sometimes require the readers to make up their own interpretations for formulae, or to keep definitions in mind that may not seem mnemonic to them.

This document is a translation of a large part of Gödel's proof. The translation happens on three levels:

- from German to English
- from Gödel's notation to more common mathematical symbols
- from paper to hyper-text

Hyper-text and colors are used as follows: definitions take place in blue italics, like this: *defined term*. Wherever the defined term is used, we have a red hyper-link to the place in the text where the term was first defined, like this: [defined term](#). Furthermore, each defined term appears in the clickable [index](#) at the end of this document. In the margin of the document, there are page-numbers like this [173], which refer to the original document. Here are links for looking up something by page: [173](#) [174](#) [175](#) [176](#) [177](#) [178](#) [179](#) [180](#) [181](#) [182](#) [183](#) [184](#) [185](#) [186](#) [187](#) [188](#) [189](#) [190](#) [191](#) [196](#). Finally, *small text snippets in magenta* are comments not present in the original text, but perhaps useful for the reader.

This translation omits all foot-notes from the original, and only contains sections [1](#) and [2](#) (out of four).

The translation comes as-is, with no explicit or implied warranty. Use at your own risk, the translator is not willing to take any responsibility for problems you might have because of errors in the translation, or because of misunderstandings. You are permitted to reproduce this document all you like, but only if you include this notice.

On formally undecidable propositions of *Principia Mathematica* and related systems I

[173]

Kurt Gödel

1931

1 Introduction

The development of mathematics towards greater exactness has, as is well-known, lead to formalization of large areas of it such that you can carry out proofs by following a few mechanical rules. The most comprehensive current formal systems are the system of *Principia Mathematica* (*PM*) on the one hand, the Zermelo-Fraenkelian axiom-system of set theory on the other hand. These two systems are so far developed that you can formalize in them all proof methods that are currently in use in mathematics, i.e. you can reduce these proof methods to a few axioms and deduction rules. Therefore, the conclusion seems plausible that these deduction rules are sufficient to decide *all* mathematical questions expressible in those systems. We will show that this is not true, but that there are even relatively easy problem in the theory of ordinary whole numbers that can not be decided from the axioms. This is not due to the nature of these systems, but it is true for a very wide class of formal systems, which in particular includes all those that you get by adding a finite number of axioms to the above mentioned systems, provided the additional axioms don't make false theorems provable.

[174]

Let us first sketch the main intuition for the proof, without going into detail and of course without claiming to be exact. The formulae of a formal system (we will restrict ourselves to the *PM* here) can be viewed syntactically as finite sequences of the basic symbols (variables, logical constants, and parentheses or separators), and it is easy to define precisely *which* sequences of the basic symbols are syntactically correct formulae and which are not. Similarly, *proofs* are formally nothing else than finite sequences of formulae (with specific definable properties). Of course, it is irrelevant for meta-mathematical observations what signs are taken for basic symbols, and so we will chose natural numbers for them. Hence, a formula is a finite sequence of natural numbers, and a proof schema is a finite sequence of finite sequences of natural numbers. The meta-mathematical concepts (theorems) hereby become concepts (theorems) about natural numbers, which makes them (at least partially) expressible in the symbols of the system *PM*. In particular, one can show that the concepts “formula”, “proof schema”, “provable formula” are all expressible within the system *PM*, i.e. one can, for example,

come up with a formula $F(v)$ of PM that has one free variable v (whose type is sequence of numbers) such that the semantic interpretation of $F(v)$ is: v is a provable formula. We will now construct an undecidable theorem of the system PM , i.e. a theorem A for which neither A nor $\neg A$ is provable, as follows:

We will call a formula of PM with exactly one free variable of type natural numbers a *class-sign*. We will assume the class-signs are somehow numbered, call the n th one R_n , and note that both the concept “class-sign” and the ordering relation R are definable within the system PM . Let α be an arbitrary class-sign; with $\alpha(n)$ we denote the formula that you get when you substitute n for the free variable of α . Also, the ternary relation $x \Leftrightarrow y(z)$ is definable within PM . Now we will define a class K of natural numbers as follows:

$$K = \{n \in \mathbb{N} \mid \neg \text{provable}(R_n(n))\} \quad (1)$$

(where $\text{provable}(x)$ means x is a provable formula). With other words, K is the set of numbers n where the formula $R_n(n)$ that you get when you insert n into its own formula R_n is improvable. Since all the concepts used for this definition are themselves definable in PM , so is the compound concept K , i.e. there is a class-sign S such that the formula $S(n)$ states that $n \in K$. As a class-sign, S is identical with a specific R_q , i.e. we have

$$S \Leftrightarrow R_q$$

for a specific natural number q . We will now prove that the theorem $R_q(q)$ is undecidable within PM . We can understand this by simply plugging in the definitions: $R_q(q) \Leftrightarrow S(q) \Leftrightarrow q \in K \Leftrightarrow \neg \text{provable}(R_q(q))$, in other words, $R_q(q)$ states “I am improvable.” Assuming the theorem $R_q(q)$ were provable, then it would also be true, i.e. because of (1) $\neg \text{provable}(R_q(q))$ would be true in contradiction to the assumption. If on the other hand $\neg R_q(q)$ were provable, then we would have $q \notin K$, i.e. $\text{provable}(R_q(q))$. That means that both $R_q(q)$ and $\neg R_q(q)$ would be provable, which again is impossible.

The analogy of this conclusion with the Richard-antinomy leaps to the eye; there is also a close kinship with the liar-antinomy, because our undecidable theorem $R_q(q)$ states that q is in K , i.e. according to (1) that $R_q(q)$ is not provable. Hence, we have in front of us a theorem that states its own unprovability. The proof method we just applied is obviously applicable to any formal system that on the one hand is expressive enough to allow the definition of the concepts used above (in particular the concept “provable formula”), and in which on the other hand all provable formulae are also true. The following exact implementation of the proof will among other things have the goal to replace the second prerequisite by a purely formal and much weaker one.

From the remark that $R_q(q)$ states its own improvability it immediately follows that $R_q(q)$ is correct, since $R_q(q)$ is in fact unprovable (because it is undecidable). The theorem which is undecidable *within the system* PM has hence been decided by meta-mathematical considerations. The exact analysis of this strange fact leads to surprising results about consistency proofs for formal systems, which will be discussed in section 4 (theorem XI).

2 Main Result

We will now exactly implement the proof sketched above, and will first give an exact description of the formal system P , for which we want to show the existence of undecidable theorems. By and large, P is the system that you get by building the logic of PM on top the Peano axioms (numbers as individuals, successor-relation as undefined basic concept).

2.1 Definitions

The *basic signs* of system P are the following:

- I. Constant: “ \neg ” (not), “ \vee ” (or), “ \forall ” (for all), “0” (zero), “*succ*” (the successor of), “(”, “)” (parentheses). Gödel’s original text uses a different notation, but the reader may be more familiar with the notation adapted in this translation.
- II. *Variable of type one* (for individuals, i.e. natural numbers including 0): “ x_1 ”, “ y_1 ”, “ z_1 ”, ...
Variables of type two (for classes of individuals, i.e. subsets of \mathbb{N}): “ x_2 ”, “ y_2 ”, “ z_2 ”, ...
Variables of type three (for classes of classes of individuals, i.e. sets of subsets of \mathbb{N}): “ x_3 ”, “ y_3 ”, “ z_3 ”, ...
 And so on for every natural number as type.

Remark: Variables for binary or n -ary functions (relations) are superfluous as **basic signs**, because one can define relations as classes of ordered pairs and ordered pairs as classes of classes, e.g. the ordered pair (a, b) by $\{\{a\}, \{a, b\}\}$, where $\{x, y\}$ and $\{x\}$ stand for the classes whose only elements are x, y and x , respectively.

By a *sign of type one* we understand a combination of signs of the form

[177]

$$a, \text{succ}(a), \text{succ}(\text{succ}(a)), \text{succ}(\text{succ}(\text{succ}(a))), \dots \text{ etc.},$$

where a is either 0 or a **variable of type one**. In the first case we call such a sign a *number-sign*. For $n > 1$ we will understand by a *sign of type n* a **variable of type n** . We call combinations of signs of the form $a(b)$, where b is a sign of type n and a a sign of type $n + 1$, *elementary formulae*. We define the class of *formulae* as the smallest set that contains all **elementary formulae** and that contains for a, b always also $\neg(a)$, $(a) \vee (b)$, $\forall x.(a)$ (where x is an arbitrary variable). We call $(a) \vee (b)$ the *disjunction* of a and b , $\neg(a)$ the *negation* and $\forall x.(a)$ the *generalization* of a . A formula that contains no free variables (where *free variables* is interpreted in the usual manner) is called *proposition-formula*. We call a formula with exactly n free individual-variables (and no other free variables) an *n -ary relation sign*, for $n = 1$ also **class-sign**.

By $\text{subst}_a \binom{v}{b}$ (where a is a **formula**, v is a **variable**, and b is a sign of the same type as v) we understand the formula that you get by substituting b for every free occurrence

of v in a . We say that a formula a is a *type-lift* of another formula b if you can obtain a from b by increasing the type of all variables occurring in a by the same number.

The following **formulae** (I through V) are called *axioms* (they are written with the help of the abbreviations (defined in the usual manner) \wedge , \Rightarrow , \Leftrightarrow , $\exists x$, $=$, and using the customary conventions for leaving out parentheses):

I. The *Peano axioms*, which give fundamental properties for natural numbers.

1. $\neg(\text{succ}(x_1) = 0)$ We start to count at 0.
2. $\text{succ}(x_1) = \text{succ}(y_1) \Rightarrow x_1 = y_1$ If two natural numbers $x_{1,2} \in \mathbb{N}$ have the same successor, they are equal.
3. $(x_2(0) \wedge \forall x_1 \cdot x_2(x_1) \Rightarrow x_2(\text{succ}(x_1))) \Rightarrow \forall x_1 \cdot x_2(x_1)$ We can prove a predicate x_2 on natural numbers by natural induction.

II. Every **formula** obtained by inserting arbitrary formulae for p, q, r in the following schemata. We call these *proposition axioms*. [178]

1. $p \vee p \Rightarrow p$
2. $p \Rightarrow p \vee q$
3. $p \vee q \Rightarrow q \vee p$
4. $(p \Rightarrow q) \Rightarrow (r \vee p \Rightarrow r \vee q)$

III. Every formula obtained from the two schemata

1. $(\forall v \cdot a) \Rightarrow \text{subst}_c^v(a)$
2. $(\forall v \cdot b \vee a) \Rightarrow (b \vee \forall v \cdot a)$

by inserting the following things for a, v, b, c (and executing the operation denoted by **subst** in 1.):

Insert an arbitrary **formula** for a , an arbitrary **variable** for v , any formula where v does not occur free for b , and for c a sign of the same type as v with the additional requirement that c does not contain a free variable that would be bound in a position in a where v is free.

For lack of a better name, we will call these *quantor axioms*.

IV. Every formula obtained from the schema

1. $\exists u \cdot \forall v \cdot (u(v) \Leftrightarrow a)$

by inserting for v and u any variables of type n and $n + 1$ respectively and for a a formula that has no free occurrence of u . This axiom takes the place of the *reducibility axiom* (the *comprehension axiom* of set theory).

V. Any formula obtained from the following by **type-lift** (and the formula itself):

$$1. (\forall x_1 \cdot (x_2(x_1) \Leftrightarrow y_2(x_1))) \Rightarrow x_2 = y_2$$

This axiom states that a class is completely determined by its elements. **Let us call it the set axiom.**

A **formula** c is called the **immediate consequence** of a and b (of a) if a is the formula $\neg b \vee c$ (or if c is the formula $\forall v \cdot a$, where v is any **variable**). The class of **provable formulae** is defined as the smallest class of **formulae** that contains the axioms and is closed under the relation “**immediate consequence**”.

2.2 Gödel-numbers

We will now uniquely associate the primitive signs of system P with natural numbers as follows:

$$\begin{array}{lll} \text{“0”} \dots 1 & \text{“succ”} \dots 3 & \text{“}\neg\text{”} \dots 5 \\ \text{“}\vee\text{”} \dots 7 & \text{“}\forall\text{”} \dots 9 & \text{“(”} \dots 11 \\ & & \text{“)”} \dots 13 \end{array}$$

[179]

Furthermore we will uniquely associate each **variable of type** n with a number of the form p^n (where p is a prime > 13). Thus there is a one-to-one-correspondence between every finite string of **basic signs** and a sequence of natural numbers. We now map the sequences of natural numbers (again in one-to-one correspondence) to natural numbers by having the sequence n_1, n_2, \dots, n_k correspond to the number $2^{n_1} \cdot 3^{n_2} \cdot \dots \cdot p_k^{n_k}$, where p_k is the k th prime (by magnitude). Thus, there is not only a uniquely associated natural number for every **basic sign**, but also for every sequence of basic signs. We will denote the number associated with the **basic sign** (resp. the sequence of basic signs) a by $\Phi(a)$. Now let $R(a_1, a_2, \dots, a_n)$ be a given class or relation between basic signs or sequences of them. We will associate that with the class (relation) $R'(x_1, x_2, \dots, x_n)$ that holds between x_1, x_2, \dots, x_n if and only if there are a_1, a_2, \dots, a_n such that for $i = 1, 2, \dots, n$ we have $x_i = \Phi(a_i)$ and the $R(a_1, a_2, \dots, a_n)$ holds. We will denote the classes and relations on natural numbers which are associated with the meta-mathematical concepts, e.g. “**variable**”, “**formula**”, “**proposition-formula**”, “**axiom**”, “**provable formula**” etc., in the above mentioned manner, by the same word in small caps. The proposition that there are undecidable problems in system P for example reads like this: There are PROPOSITION-FORMULAE a , such that neither a nor the NEGATION of a is a PROVABLE FORMULA.

2.3 Primitive recursion

At this point, we will make an excursion to make an observation that *a priori* does not have anything to do with the system P , and will first give the following definition: we

say a number-theoretical **formula** $\phi(x_1, x_2, \dots, x_n)$ is *defined via primitive recursion* in terms of the number-theoretical formulae $\psi(x_1, x_2, \dots, x_{n-1})$ and $\mu(x_1, x_2, \dots, x_{n+1})$ if the following holds for all x_2, \dots, x_n, k :

$$\begin{aligned}\phi(0, x_2, \dots, x_n) &= \psi(x_2, \dots, x_n), \\ \phi(k+1, x_2, \dots, x_n) &= \mu(k, \phi(k, x_2, \dots, x_n), x_2, \dots, x_n)\end{aligned}\tag{2}$$

We call a number-theoretical formula ϕ *primitive recursive* if there is a finite sequence of number-theoretical formulae $\phi_1, \phi_2, \dots, \phi_n$ ending in ϕ such that every function ϕ_k of the sequence is either defined from two of the preceding formulae by **primitive recursion** or results by inserting into any of the preceding ones or, and this is the base case, is a constant or the successor function $\text{succ}(x) = x + 1$. The length of the shortest sequence of ϕ_i belonging to a **primitive recursive** function ϕ is called its *degree*. We call a relation $R(x_1, \dots, x_n)$ *primitive recursive* if there is a **primitive recursive** function $\phi(x_1, \dots, x_n)$ such that for all x_1, x_2, \dots, x_n ,

$$R(x_1, \dots, x_n) \Leftrightarrow (\phi(x_1, \dots, x_n) = 0).$$

The following theorems hold:

- I. *Every function (relation) that you get by inserting **primitive recursive** functions in the places of variables of other **primitive recursive** functions (relations) is itself **primitive recursive**; likewise every function that you get from **primitive recursive** functions by the schema (2).*
- II. *If R and S are **primitive recursive** relations, then so are $\neg R, R \vee S$ (and therefore also $R \wedge S$).*
- III. *If the functions $\phi(\vec{x}), \psi(\vec{y})$ are **primitive recursive**, then so is the relation $\phi(\vec{x}) = \psi(\vec{y})$. *We have resorted to a vector notation \vec{x} to denote finite-length tuples of variables.**
- IV. *If the function $\phi(\vec{x})$ and the relation $R(y, \vec{z})$ are **primitive recursive**, then so are the relations S, T*

$$S(\vec{x}, \vec{z}) \Leftrightarrow (\exists y \leq \phi(\vec{x}) . R(y, \vec{z}))$$

$$T(\vec{x}, \vec{z}) \Leftrightarrow (\forall y \leq \phi(\vec{x}) . R(y, \vec{z}))$$

as well as the function ψ

$$\psi(\vec{x}, \vec{z}) = (\mathbf{argmin} y \leq \phi(\vec{x}) . R(y, \vec{z}))$$

where **argmin** $x \leq f(x) . F(x)$ stands for the smallest x for which $(x \leq f(x)) \wedge F(x)$ holds, and for 0 if there is no such number. *Readers to whom an operational*

[180]

description appeals more may want to think of this as a loop that tries every value from 1 to $\phi(\vec{x})$ to determine the result. The crucial point here is this theorem does not state that an unbounded loop (or recursion) is primitive recursive; those are in fact strictly more powerful in terms of computability.

Theorem **I** follows immediately from the definition of “**primitive recursive**”. Theorems **II** and **III** are based upon the fact that the number-theoretical functions

$$\alpha(x), \beta(x, y), \gamma(x, y)$$

corresponding to the logical concepts $\neg, \vee, =$ (where $n = 0$ is taken for **true** and $n \neq 0$ for **false**), namely

$$\begin{aligned} \alpha(x) &= \begin{cases} 1 & \text{for } x = 0 \\ 0 & \text{for } x \neq 0 \end{cases} \\ \beta(x, y) &= \begin{cases} 0 & \text{if one or both of } x, y \text{ are } = 0 \\ 1 & \text{if both } x, y \text{ are } \neq 0 \end{cases} \\ \gamma(x, y) &= \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases} \end{aligned}$$

are **primitive recursive**, as one can easily convince oneself. The proof for theorem **IV** is, in short, the following: by assumption there is a **primitive recursive** $\rho(y, \vec{z})$ such that: [181]

$$R(y, \vec{z}) \Leftrightarrow (\rho(y, \vec{z}) = 0)$$

Using the recursion-schema (2) we now define a function $\chi(y, \vec{z})$ as follows:

$$\begin{aligned} \chi(0, \vec{z}) &= 0 \\ \chi(n+1, \vec{z}) &= (n+1) \cdot A + \chi(n, \vec{z}) \cdot \alpha(A) \end{aligned}$$

where $A = \alpha(\alpha(\rho(0, \vec{z}))) \cdot \alpha(\rho(n+1, \vec{z})) \cdot \alpha(\chi(n, \vec{z}))$.

A , which makes use of the above defined α and of the fact that a product is 0 if one of its factors is 0, can be described by the following pseudo-code:

```
A = if( $\rho(0, \vec{z}) = 0$ )
  then 0
  else if( $\rho(n+1, \vec{z}) \neq 0$ )
    then 0
    else if( $\chi(n, \vec{z}) \neq 0$ )
      then 0
      else 1
```

It is a nice example for how arithmetic can be used to emulate logics.

Therefore, $\chi(n+1, \vec{z})$ is either $= n+1$ (if $A = 1$) or $= \chi(n, \vec{z})$ (if $A = 0$). Obviously, the first case will occur if and only if all factors of A are 1, i.e. if we have

$$\neg R(0, \vec{z}) \wedge R(n+1, \vec{z}) \wedge (\chi(n, \vec{z}) = 0).$$

This implies that the function $\chi(n, \vec{z})$ (viewed as a function of n) remains 0 up to the smallest value of n for which $R(n, \vec{z})$ holds, and has that value from then on (if $R(0, \vec{z})$ already holds then $\chi(n, \vec{z})$ is correspondingly constant and = 0). Therefore, we have

$$\begin{aligned} \psi(\vec{x}, \vec{z}) &= \chi(\phi(\vec{x}), \vec{z}) \\ S(\vec{x}, \vec{z}) &\Leftrightarrow R(\psi(\vec{x}, \vec{z}), \vec{z}). \end{aligned}$$

It is easy to reduce the relation T to a case analogous to that of S by negation. This concludes the proof of theorem IV.

2.4 Expressing metamathematical concepts

As one can easily convince oneself, the functions $x + y$, $x \cdot y$, x^y and furthermore the relations $x < y$ and $x = y$ are **primitive recursive**. For example, the function $x + y$ can be constructed as $0 + y = y$ and $(k + 1) + y = \text{succ}(k + y)$, i.e. $\psi(y) = y$ and $\mu(k, l, y) = \text{succ}(l)$ in schema (2). Using these concepts, we will now define a sequence of functions (relations) 1-45, each of which is defined from the preceding ones by the methods given by theorems I through IV. In doing so, usually multiple of the definition steps allowed by theorems I through IV are combined in one. Each of the functions (relations) 1-45, among which we find for example the concepts “FORMULA”, “AXIOM”, and “IMMEDIATE CONSEQUENCE”, is therefore **primitive recursive**.

[182]

1. $y \mid x \Leftrightarrow \exists z \leq x . x = y \cdot z$
 x is divisible by y .
2. $\text{isPrime}(x) \Leftrightarrow \neg(\exists z \leq x . (z \neq 1 \wedge z \neq x \wedge z \mid x)) \wedge (x > 1)$
 x is a prime number.
3. $\text{prFactor}(0, x) = 0$
 $\text{prFactor}(n+1, x) = \mathbf{argmin} y \leq x . (\text{isPrime}(y) \wedge y \mid x \wedge y > \text{prFactor}(n, x))$
 $\text{prFactor}(n, x)$ is the n th (by size) prime number contained in x .
4. $0! = 1$
 $(n+1)! = (n+1) \cdot n!$
5. $\text{nthPrime}(0) = 0$
 $\text{nthPrime}(n+1) = \mathbf{argmin} y \leq (\text{nthPrime}(n)! + 1) . (\text{isPrime}(y) \wedge y > \text{nthPrime}(n))$
 $\text{nthPrime}(n)$ is the n th (by size) prime number.

6. $item(n, x) = \mathbf{argmin} y \leq x . ((prFactor(n, x)^y \mid x) \wedge \neg(prFactor(n, x)^{y+1} \mid x))$
 $item(n, x)$ is the n th item of the sequence of numbers associated with x (for $n > 0$ and n not larger than the length of this sequence).

7. $length(x) = \mathbf{argmin} y \leq x . (prFactor(y, x) > 0 \wedge prFactor(y + 1, x) = 0)$
 $length(x)$ is the length of the sequence of numbers associated with x .

8. $x \circ y = \mathbf{argmin} z \leq nthPrime(length(x) + length(y))^{x+y} .$
 $(\forall n \leq length(x) . item(n, z) = item(n, x)) \wedge$
 $(\forall 0 < n \leq length(y) . item(n + length(x), z) = item(n, y))$
 $x \circ y$ corresponds to the operation of “concatenating” two finite sequences of numbers.

9. $seq(x) = 2^x$
 $seq(x)$ corresponds to the number sequence that consists only of the number x (for $x > 0$).

10. $paren(x) = seq(11) \circ x \circ seq(13)$
 $paren(x)$ corresponds to the operation of “parenthesizing” (11 and 13 are associated with the primitive signs “(” and “”).

11. $vtype(n, x) \Leftrightarrow (\exists 13 < z \leq x . isPrime(z) \wedge x = z^n) \wedge n \neq 0$
 x is a **VARIABLE OF TYPE n** .

12. $isVar(x) \Leftrightarrow \exists n \leq x . vtype(n, x)$
 x is a **VARIABLE**.

13. $not(x) = seq(5) \circ paren(x)$
 $not(x)$ is the **NEGATION** of x .

[183]

14. $or(x, y) = paren(x) \circ seq(7) \circ paren(y)$
 $or(x, y)$ is the **DISJUNCTION** of x and y .

15. $forall(x, y) = seq(9) \circ seq(x) \circ paren(y)$
 $forall(x, y)$ is the **GENERALIZATION** of y by the **VARIABLE** x (provided that x is a **VARIABLE**).

16. $succ_n(0, x) = x$
 $succ_n(n + 1, x) = seq(3) \circ succ_n(n, x)$
 $succ_n(n, x)$ corresponds to the operation of “prepending the sign ‘succ’ in front of x for n times”.

17. $number(n) = succ(n, seq(1))$
 $number(n)$ is the **NUMBER-SIGN** for the number n .

18. $stype_1(x) \Leftrightarrow \exists m, n \leq x .$
 $(m = 1 \vee vtype(1, m)) \wedge x = succ_n(n, seq(m))$
 x is a **SIGN OF TYPE ONE**.

$$19. \text{stype}(n, x) \Leftrightarrow \begin{cases} n = 1 \wedge \text{stype}_1(x) \\ n > 1 \wedge \exists v \leq x. (\text{vtype}(n, v) \wedge x = R(v)) \end{cases}$$

x is a **SIGN OF TYPE** n .

$$20. \text{elFm}(x) \Leftrightarrow \exists y, z, n \leq x. (\text{stype}(n, y) \wedge \text{stype}(n+1, z) \wedge x = z \circ \text{paren}(y))$$

x is an **ELEMENTARY FORMULA**.

$$21. \text{op}(x, y, z) \Leftrightarrow (x = \text{not}(y)) \vee (x = \text{or}(y, z)) \vee (\exists v \leq x. \text{isVar}(v) \wedge x = \text{forall}(v, y))$$

$$22. \text{fmSeq}(x) \Leftrightarrow (\forall 0 < n \leq \text{length}(x). \text{elFm}(\text{item}(n, x))) \vee \\ \exists 0 < p, q < n. \text{op}(\text{item}(n, x), \text{item}(p, x), \text{item}(q, x))) \\ \wedge \text{length}(x) > 0$$

x is a sequence of **FORMULAE**, each of which is either an **ELEMENTARY FORMULA** or is obtained from the preceding ones by the operations of **NEGATION**, **DISJUNCTION**, or **GENERALIZATION**.

$$23. \text{isFm}(x) \Leftrightarrow \exists n \leq \left(\text{nthPrime}(\text{length}(x)^2) \right)^{x \cdot (\text{length}(x))^2}. \\ \text{fmSeq}(n) \wedge x = \text{item}(\text{length}(n), n)$$

x is a **FORMULA** (i.e. the last item of a sequence n of formulae).

$$24. \text{bound}(v, n, x) \Leftrightarrow \text{isVar}(v) \wedge \text{isFm}(x) \wedge \\ \exists a, b, c \leq x. x = a \circ \text{forall}(v, b) \circ c \wedge \text{isFm}(b) \wedge \\ \text{length}(a) + 1 \leq n \leq \text{length}(a) + \text{length}(\text{forall}(v, b))$$

The variable v is **BOUND** in x at position n .

[184]

$$25. \text{free}(v, n, x) \Leftrightarrow \text{isVar}(v) \wedge \text{isFm}(x) \wedge \\ v = \text{item}(n, x) \wedge n \leq \text{length}(x) \wedge \neg \text{bound}(v, n, x)$$

The variable v is **FREE** in x at position n .

$$26. \text{free}(v, x) \Leftrightarrow \exists n \leq \text{length}(x). \text{free}(v, n, x)$$

v occurs in x as a **FREE VARIABLE**.

$$27. \text{insert}(x, n, y) = \mathbf{argmin} z \leq (\text{nthPrime}(\text{length}(x) + \text{length}(y)))^{x+y}. \\ \exists u, v \leq x.$$

$$x = u \circ \text{seq}(\text{item}(n, x)) \circ v \wedge z = u \circ y \circ v \wedge n = \text{length}(u) + 1$$

You obtain $\text{insert}(x, n, y)$ from x by inserting y instead of the n th item in the sequence x (provided that $0 < n \leq \text{length}(x)$).

$$28. \text{freePlace}(0, v, x) = \mathbf{argmin} n \leq \text{length}(x). \\ \text{free}(v, n, x) \wedge \neg \exists n < p \leq \text{length}(x). \text{free}(v, p, x) \\ \text{freePlace}(k+1, v, x) = \mathbf{argmin} n < \text{freePlace}(n, k, v). \\ \text{free}(v, n, x) \wedge \neg \exists n < p < \text{freePlace}(n, k, v). \text{free}(v, p, x)$$

$\text{freePlace}(k, v, x)$ is the $k+1$ st place in x (counted from the end of **FORMULA** x) where v is **FREE** (and 0 if there is no such place).

29. $nFreePlaces(v, x) = \mathbf{argmin} n \leq length(x) . freePlace(n, v, x) = 0$
 $nFreePlaces(v, x)$ is the number of places where v is free in x .

30. $subst'(0, x, v, y) = x$
 $subst'(k + 1, x, v, y) = insert(subst'(k, x, v, y), freePlace(k, v, x), y)$

31. $subst(x, v, y) = subst'(nFreePlaces(v, x), x, v, y)$
 $subst(x, v, y)$ is the above defined concept $\mathbf{substa} \binom{v}{b}$.

32. $imp(x, y) = or(not(x), y)$
 $and(x, y) = not(or(not(x), not(y)))$
 $equiv(x, y) = and(imp(x, y), imp(y, x))$
 $exists(v, y) = not(forall(v, not(y)))$

33. $typeLift(n, x) = \mathbf{argmin} y \leq x^{x^n} .$
 $\forall k \leq length(x) .$

$item(k, x) \leq 13 \wedge item(k, y) = item(k, x) \vee$
 $item(k, x) > 13 \wedge item(k, y) = item(k, x) \cdot prFactor(1, item(k, x))^n$

$typeLift(n, x)$ is the n th **TYPE-LIFT** of x (if x and $typeLift(n, x)$ are **FORMULAE**).

There are three specific numbers corresponding to the axioms I, 1 to 3 (**the Peano axioms**), which we will denote by pa_1, pa_2, pa_3 , and we define:

34. $peanoAxiom(x) \Leftrightarrow (x = pa_1 \vee x = pa_2 \vee x = pa_3)$

[185]

35. $prop1Axiom(x) \Leftrightarrow \exists y \leq x . isFm(y) \wedge x = imp(or(y, y), y)$

x is a **FORMULA** that has been obtained by inserting into the axiom schema II, 1. We define $prop2Axiom(x)$, $prop3Axiom(x)$, and $prop4Axiom(x)$ analogously.

36. $propAxiom(x) \Leftrightarrow prop1Axiom(x) \vee prop2Axiom(x) \vee prop3Axiom(x) \vee prop4Axiom(x)$
 x is a **FORMULA** that has been obtained by inserting into on of the **proposition axioms**.

37. $quantor1AxiomCondition(z, y, v) \Leftrightarrow \neg \exists n \leq length(y), m \leq length(z), w \leq z .$
 $w = item(m, z) \wedge bound(w, n, y) \wedge free(v, n, y)$

z does not contain a **VARIABLE** that is **BOUND** anywhere in y where v is **FREE**. **This condition for the applicability of axiom III, 1, ensured that a substitution of z for the free occurrences of v in y does not accidentally bind some of z 's variables.**

38. $quantor1Axiom(x) \Leftrightarrow \exists v, y, z, n \leq x .$
 $vtype(n, v) \wedge stype(n, z) \wedge isFm(y) \wedge quantor1AxiomCondition(z, y, v) \wedge$
 $x = imp(forall(v, y), subst(y, v, z))$

x is a **FORMULA** obtained by substitution from the axiom schema III, 1, i.e. **one of the quantor axioms**.

39. $quantor2Axiom(x) \Leftrightarrow \exists v, q, p \leq x .$
 $isVar(v) \wedge isFm(p) \wedge \neg free(v, p) \wedge isFm(q) \wedge$

$$x = \text{imp}(\text{forall}(v, \text{or}(p, q)), \text{or}(p, \text{forall}(v, q)))$$

x is a **FORMULA** obtained by substitution from the axiom schema III, 2, i.e. the other one of the quantor axioms.

$$40. \text{reduAxiom}(x) \Leftrightarrow \exists u, v, y, n \leq x . \\ \text{vtype}(n, v) \wedge \text{vtype}(n + 1, u) \wedge \neg \text{free}(u, y) \wedge \text{isFm}(y) \wedge \\ x = \text{exists}(u, \text{forall}(v, \text{equiv}(\text{seq}(u) \circ \text{paren}(\text{seq}(v)), y)))$$

x is a **FORMULA** obtained by substitution from the axiom schema IV, 1, i.e. from the reducibility axiom.

There is a specific number corresponding to axiom V, 1, (the set axiom), which we will denote by sa , and we define:

$$41. \text{setAxiom}(x) \Leftrightarrow \exists n \leq x . x = \text{typeLift}(n, sa)$$

$$42. \text{isAxiom}(x) \Leftrightarrow \text{peanoAxiom}(x) \vee \text{propAxiom}(x) \vee \\ \text{quantor1Axiom}(x) \vee \text{quantor2Axiom}(x) \vee \text{reduAxiom}(x) \vee \\ \text{setAxiom}(x)$$

x is an **AXIOM**.

$$43. \text{immConseq}(x, y, z) \Leftrightarrow y = \text{imp}(z, x) \vee \exists v \leq x . \text{isVar}(v) \wedge x = \text{forall}(v, y)$$

x is an **IMMEDIATE CONSEQUENCE** of y and z .

$$44. \text{isProofFigure}(x) \Leftrightarrow (\forall 0 < n \leq \text{length}(x) . \\ \text{isAxiom}(\text{item}(n, x)) \vee \exists 0 < p, q < n . \\ \text{immConseq}(\text{item}(n, x), \text{item}(p, x), \text{item}(q, x))) \wedge \\ \text{length}(x) > 0$$

x is a **PROOF FIGURE** (a finite sequence of **FORMULAE**, each of which is either an **AXIOM** or the **IMMEDIATE CONSEQUENCE** of two of the preceding ones).

$$45. \text{proofFor}(x, y) \Leftrightarrow \text{isProofFigure}(x) \wedge \text{item}(\text{length}(x), x) = y$$

x is a **PROOF** for the **FORMULA** y .

$$46. \text{provable}(x) \Leftrightarrow \exists y . \text{proofFor}(y, x)$$

x is a **PROVABLE FORMULA**. ($\text{provable}(x)$ is the only one among the concepts 1-46 for which we can not assert that it is **primitive recursive**).

2.5 Denotability and provability

The fact that can be expressed vaguely by: Every **primitive recursive** relation is definable within system P (interpreting that system as to content), will be expressed in the following theorem *without* referring to the interpretation of **formulae** of P :

Theorem V: For every **primitive recursive** relation $R(x_1, \dots, x_n)$ there is a **RELATION SIGN** r (with the **FREE VARIABLES** u_1, \dots, u_n), such that for each n -tuple (x_1, \dots, x_n) the following holds:

$$R(x_1, \dots, x_n) \Rightarrow \text{provable}(\text{subst}(r, u_1 \dots u_n, \text{number}(x_1) \dots \text{number}(x_n))) \quad (3)$$

$$\neg R(x_1, \dots, x_n) \Rightarrow \text{provable}(\text{not}(\text{subst}(r, u_1 \dots u_n, \text{number}(x_1) \dots \text{number}(x_n)))) \quad (4)$$

We contend ourselves with giving a sketchy outline of the proof for this theorem here, since it does not offer any difficulties in principle and is rather cumbersome. We prove the theorem for all relations $R(x_1, \dots, x_n)$ of the form $x_1 = \phi(x_2, \dots, x_n)$ (where ϕ is a **primitive recursive** function) and apply natural induction by ϕ 's **degree**. For functions of **degree** one (i.e. constants and the function $x + 1$) the theorem is trivial. Hence, let ϕ be of **degree** m . It is built from functions of lower degree ϕ_1, \dots, ϕ_k by the operations of insertion and **primitive recursive** definition. Since everything has already been proven for ϕ_1, \dots, ϕ_k by the inductive assumption, there are corresponding RELATION SIGNS r_1, \dots, r_k such that (3), (4) hold. The definition processes by which ϕ is built from ϕ_1, \dots, ϕ_k (insertion and primitive recursion) can all be modeled formally in system P . Doing this, one gets from r_1, \dots, r_k a new RELATION SIGN r for which one can proof the validity of (3), (4) without difficulties. A RELATION SIGN r associated with a primitive recursive relation in this manner shall be called *primitive recursive*. [187]

2.6 Undecidability theorem

We now come to the goal of our elaborations. Let κ be any class of **FORMULAE**. We denote with $\text{Conseq}(\kappa)$ the smallest set of **FORMULAE** that contains all **FORMULAE** of κ and all **AXIOMS** and is closed under the relation "IMMEDIATE CONSEQUENCE". κ is called *ω -consistent* if there is no **CLASS-SIGN** a such that

$$(\forall n. \text{subst}(a, v, \text{number}(n)) \in \text{Conseq}(\kappa)) \wedge \text{not}(\text{forall}(v, a)) \in \text{Conseq}(\kappa)$$

where v is the **FREE VARIABLE** of the **CLASS-SIGN** a . With other words, a witness against ω -consistency would be a formula a with one free variable where we can derive $a(n)$ for all n , but also $\neg \forall n. a(n)$, a contradiction.

Every ω -consistent system is, of course, also consistent. The reverse, however, does not hold true, as will be shown later. We call a system *consistent* if there is no formula a such that both a and $\neg a$ are provable. Such a formula would be a witness against the consistency, but in general not against the ω -consistency. With other words, ω -consistency is stronger than consistency: the first implies the latter, but not vice versa.

The general result about the existence of undecidable propositions goes as follows:

Theorem VI: *For every ω -consistent primitive recursive class κ of **FORMULAE** there is a primitive recursive **CLASS-SIGN** r such that neither $\text{forall}(v, r)$ nor $\text{not}(\text{forall}(v, r))$ belongs to $\text{Conseq}(\kappa)$ (where v is the **FREE VARIABLE** of r).*

Since the premise in the theorem is ω -consistency, which is stronger than consistency, the theorem is less general than if its premise were just consistency.

Proof: Let κ be any ω -consistent primitive recursive class of FORMULAE. We define:

$$\begin{aligned} isProofFigure_{\kappa}(x) \Leftrightarrow & \\ & (\forall n \leq length(x) . isAxiom(item(n, x)) \vee (item(n, x) \in \kappa) \vee \\ & \exists 0 < p, q < n . immedConseq(item(n, x), item(p, x), item(q, x))) \wedge \\ & length(x) > 0 \end{aligned} \quad (5)$$

(compare to the analogous concept 44)

$$proofFor_{\kappa}(x, y) \Leftrightarrow isProofFigure_{\kappa}(x) \wedge item(length(x), x) = y \quad (6)$$

$$provable_{\kappa}(x) \Leftrightarrow \exists y . proofFor_{\kappa}(y, x) \quad (6.1)$$

(compare to the analogous concepts 45, 46).

The following obviously holds:

$$\forall x . (provable_{\kappa}(x) \Leftrightarrow x \in Conseq(\kappa)), \quad (7)$$

$$\forall x . (provable(x) \Rightarrow provable_{\kappa}(x)). \quad (8)$$

Now we define the relation:

[188]

$$Q(x, y) \Leftrightarrow \neg (proofFor_{\kappa}(x, subst(y, 19, number(y))).) \quad (8.1)$$

Intuitively $Q(x, y)$ means x does not prove $y(y)$.

Since $proofFor_{\kappa}(x, y)$ (by (6), (5)) and $subst(y, 19, number(y))$ (by definitions 17, 31) are primitive recursive, so is $Q(x, y)$. According to theorem V we hence have a RELATION SIGN q (with the FREE VARIABLES 17, 19) such that the following holds:

$$\neg proofFor_{\kappa}(x, subst(y, 19, number(y))) \Rightarrow provable_{\kappa}(subst(q, 17, 19, number(x), number(y))) \quad (9)$$

$$proofFor_{\kappa}(x, subst(y, 19, number(y))) \Rightarrow provable_{\kappa}(not(subst(q, 17, 19, number(x), number(y)))) \quad (10)$$

We set:

$$p = forall(17, q) \quad (11)$$

(p is a CLASS-SIGN with the FREE VARIABLE 19 (which intuitively means 19(19), i.e. $y(y)$, is improvable)) and

$$r = subst(q, 19, number(p)) \quad (12)$$

(r is a **primitive recursive CLASS-SIGN** with the **FREE VARIABLE 17** (which intuitively means that 17, i.e. x , does not prove $p(p)$, where $p(p)$ means $p(p)$ is unprovable)).

Then the following holds:

$$\begin{aligned} subst(p, 19, number(p)) &= subst(forall(17, q), 19, number(p)) \\ &= forall(17, subst(q, 19, number(p))) \\ &= forall(17, r) \end{aligned} \quad (13)$$

(because of (11 and 12)); furthermore:

$$subst(q, 17, 19, number(x) \ number(p)) = subst(r, 17, number(x)) \quad (14)$$

(because of (14)). The recurring $forall(17, r)$ can be interpreted as there is no prove for $p(p)$, with other words, $forall(17, r)$ states that the statement $p(p)$ that states its own improbability is improvable. If we now insert p for y in (9) and (10), we get, taking (13) and (14) into account:

$$\neg proofFor_{\kappa}(x, forall(17, r)) \Rightarrow provable_{\kappa}(subst(r, 17, number(x))) \quad (15)$$

$$proofFor_{\kappa}(x, forall(17, r)) \Rightarrow provable_{\kappa}(not(subst(r, 17, number(x)))) \quad (16)$$

This yields:

[189]

1. $forall(17, r)$ is not κ -PROVABLE. Because if that were the case, there would (by (7)) exist an n such that $proofFor_{\kappa}(n, forall(17, r))$. By (16) we would hence have:

$$provable_{\kappa}(not(subst(r, 17, number(n)))),$$

while on the other hand the κ -PROVABILITY of $forall(17, r)$ also implies that of $subst(r, 17, number(n))$. Therefore κ would be inconsistent (and in particular ω -inconsistent).

2. $not(forall(17, r))$ is not κ -PROVABLE. Proof: As has just been shown, $forall(17, r)$ is not κ -PROVABLE, i.e. (by (7)) we have

$$\forall n . \neg proofFor_{\kappa}(n, forall(17, r)).$$

This implies by (15)

$$\forall n . provable_{\kappa}(subst(r, 17, number(n)))$$

which would, together with

$$provable_{\kappa}(not(forall(17, r))),$$

contradict the ω -consistency of κ .

Therefore $forall(17, r)$ is not decidable from κ , whereby theorem VI is proved.

2.7 Discussion

One can easily convince oneself that the proof we just did is constructive, i.e. it the following is intuitionistically flawlessly proven:

Let any **primitive recursively** defined class κ of **formulae** be given. Then if the formal decision (from κ) of the **PROPOSITION-FORMULA** $forall(17, r)$ is also given, one can effectively present:

1. A PROOF for $not(forall(17, r))$.
2. For any given n a PROOF for $subst(r, 17, number(n))$, i.e. a formal decision for $forall(17, r)$ would imply the effective presentability of an ω -inconsistency-proof.

Let us call a relation (class) between natural numbers $R(x_1, \dots, x_n)$ *decision-definite* if there is an n -ary RELATION SIGN r such that (3) and (4) (c.f. theorem V) hold. In particular therefore every **primitive recursive** relation is by Theorem V **decision-definite**. Analogously, a RELATION SIGN shall be called *decision-definite* if it corresponds to a **decision-definite** relation in this manner. For the existence of propositions undecidable from κ it is now sufficient to require of a class κ that it is **ω -consistent** and **decision-definite**. **With other words, it is not even important how the class of added axioms κ is defined, we just have to be able to decide with the means of the system whether something is an axiom or not.** This is because the **decision-definiteness** carries over from κ to $proofFor_\kappa(x, y)$ (compare to (5), (6)) and to $Q(x, y)$ (compare to (9)), and only that was used for the above proof. In this case, the undecidable theorem takes on the form $forall(v, r)$, where r is a **decision-definite CLASS-SIGN** (by the way, it is even sufficient that κ is **decision-definite** in the system augmented by κ).

[190]

If instead of **ω -consistency** we only assume **consistency** for κ , then, although the existence of an undecidable proposition does not follow, there follows the existence of a property (r) for which a counter-example is not *presentable* and neither is it provable that the relation holds for all numbers. Because for the proof that $forall(17, r)$ is not ω -PROVABLE we only used the ω -consistency of κ (compare to page 189), and $\neg provable_\kappa(forall(17, r))$ implies by (15) for each number x that $subst(r, 17, number(x))$ holds, i.e. that for no number $not(subst(r, 17, number(x)))$ is provable.

If you add $not(forall(17, r))$ to κ you get a consistent but not ω -consistent class of **FORMULAE** κ' . κ' is consistent because otherwise $forall(17, r)$ would be provable. But κ' is not ω -consistent, since because of $\neg provable_\kappa(forall(17, r))$ and (15) we have

$$\forall x . provable_\kappa(subst(r, 17, number(x))),$$

and hence in particular

$$\forall x . provable_{\kappa'}(subst(r, 17, number(x))),$$

and on the other hand of course

$$\text{provable}_{\kappa'}(\neg \text{forall}(17, r)).$$

But that means that $\text{forall}(17, r)$ precisely fits the definition of a witness against ω -consistency.

A special case of theorem VI is the theorem where the class κ consists of a finite number of FORMULAE (and perhaps the ones derived from these by TYPE-LIFT). Every finite class κ is of course primitive recursive. Let a be the largest contained number. Then we have for κ in this case

$$x \in \kappa \Leftrightarrow \exists m \leq x, n \leq a. n \in \kappa \wedge x = \text{typeLift}(m, n)$$

Hence, κ is primitive recursive. This allows us to conclude for example that also with the help of the axiom of choice (for all types) or the generalized continuum hypothesis not all propositions are decidable, assuming that these hypotheses are ω -consistent.

During the proof of theorem VI we did not use any other properties of the system P than the following:

1. The class of axioms and deduction rules (i.e. the relation “immediate consequence”) are primitive recursively definable (as soon as you replace the basic signs by numbers in some way).
2. Every primitive recursive relation is definable within the system P (in the sense of theorem V).

Hence there are undecidable propositions of the form $\forall x. F(x)$ in every formal system that fulfills the preconditions 1, 2 and is ω -consistent, and also in every extension of such a system by a primitive recursively definable, ω -consistent class of axioms. To this kind of systems belong, as one can easily confirm, the Zermelo-Fraenkelian axiom-system and the von Neumannian system of set-theory, furthermore the axiom-system of number-theory which consists of the Peano axioms, primitive recursive definition (by schema (2)) and the logical deduction rules. Simply every system whose deduction rules are the usual ones and whose axioms (analogously like in P) are made by insertion into a finite number of schemas fulfills precondition 1. [191]

3 Generalizations

—omitted—

[196]

4 Implications for the nature of consistency

—omitted—

A Experiences

This translation was done for a reason. I took Mike Eisenberg’s class “Computer Science: The Canon” at the University of Colorado in fall 2000. It was announced as a “great works” lecture-and-discussion course, offering an opportunity to be pointed to some great papers, giving an incentive to read them, and providing a forum for discussion. This also explains my motivation for translating Gödel’s proof: it is a truly impressive and fascinating paper, there is some incentive in completing my final paper for a class, and this exercise should and did benefit me intellectually. Here, I will try to share the experiences I made doing the translation. I deliberately chose a personal, informal style for this final section to stress that what I write here are just my opinions, nothing less and nothing more.

A.1 Have I learned or gained something?

First of all, how useful is it to read this paper anyway, whether you translate it or not? One first answer that comes to mind is that the effort of understanding it hones abstract thinking skills, and that some basic concepts like **Peano axioms**, **primitive recursion**, or **consistency** are nicely illustrated and shown in a motivated context. The difficulty with this argument is that it is self-referential: we read this paper to hone skills that we would not need to hone if we would not read this kind of papers. I don’t really have a problem with that, people do many things for their own sake, but fortunately there are other gains to be had from reading this paper. One thing that I found striking is how I only fully appreciated the thoughts from section 2.7 on re-reading the proof. I find it fascinating just how general the result is: your formal system does not need to be finite, or even primitive recursively describable, no, it suffices that you can decide its set of axioms in itself. I am not sure whether other people are as fascinated by this as I am; if you are not, try to see what I mean, it’s worth the effort! But in any case, I have gained an appreciation for the beauty and power of the results, which I am happy for. Finally, there is some hope that the writing skills, proof techniques, and thought processes exhibited by this paper might rub off, so to speak. Part of learning an art is to study the masters, and Gödel was clearly a master in his art!

Second, how useful was the translation itself? Well, it was useful to try out some ideas I had about how translating a technical paper between languages might work. My recipe was to first read and understand the whole paper, then translate it one sentence at a time (avoid to start translating a sentence before having a plan for all of it!), and finally to read it fast to check the flow and logic. This might or might not be the best way, but it worked well enough for me. For understanding the paper itself, the translation between languages or the use of hyper-text as a medium did not help me much as I was doing it. More important was the translation of notation to one I am more used to, and the occasional comment to express my view of a tricky detail. Last but not least, it seems hardly necessary to admit my strong affection for type-setting, and there is a certain pleasure in looking over and polishing something you crafted that

I believe I share with many people.

A.2 Has the paper improved?

The original paper is brilliant, well-written, rich of content, relevant. Yet I went ahead and tinkered around, changing a little thing here and there, taking much more freedom than the translators for [2, 1]. Yes, the paper did improve! It is closer to my very personal ideas of what it should ideally look like. To me who did the changes just a few days ago the modified paper looks better than the original.

In section 0, I announced a translation along three dimensions, namely language (from German to English), notation (using symbols I am more used to) and medium (exploiting hyper-text). Let us review each one in turn and criticize the changes.

Language. After finishing the translation, I compared it with the ones in [2, 1], and found that although they are different, the wording probably does not matter all that much. To give an example where it did seem to play a role, here are three wordings for *besteht eine nahe Verwandtschaft*: (i) *is closely related*, (ii) *is also a close relationship*, (iii) *is also a close kinship*. The third one is mine, and my motivation for it is that it is the most punchy one, for what it's worth. The stumbling blocks in this kind of translation, as I see it, are rather the technical terms that may be in no dictionary. For example, I could well imagine that my translation *decision-definite* seems unnatural to someone studying logic who might be used to another term.

Notation. This is the part of the translation that I believe helps the most in making the paper more accessible for readers with a similar educational background as mine. For example, I have never seen “ Π ” used for “ \forall ”, and inside an English text I find “*bound*(v, n, x)” easier to parse than “ v Geb n, x ”.

Hyper-text. During the translation, it became clearer to me just how very “hyper-text” the paper already was! On the one hand, the fact that one naturally refers back to definitions and theorems underlines that hyper-text is a natural way of presentation. On the other hand, the fact that one gets along quite well with a linear text, relying on the readers to construct the thought-building in their own head, seems to suggest that the change of medium was in fact rather superficial. I would be interested in the opinions of readers of this document on this: did the hyper-text improve the paper?

Clearly, the most important aspects of the paper are still the organization, writing, and explanation skills of Gödel himself. And clearly, the paper is still an intellectual challenge, yielding its rewards only to the fearless. To assume that my work has changed either of these facts significantly would be presumptuous.

A.3 Opinions

This discussion is a trade-off between being careful and thoughtful on the one hand, and being forthcoming and fruitful on the other. As it leans more to the second half of the spectrum, I might as well go ahead and state some opinions the project and the reflections upon it have inspired in me.

- A well-written technical paper already has the positive features of hyper-text. This may not seem so at first glance, but compare it to the typical web-page and then ask yourself which has more coherence. To me, coherence is part of the essence and beauty of cross-referencing.
- There is an analogy between writing papers and computer programs, and it is amazing how far you can stretch it without it breaks down. The skill of gradually building up your vocabulary, dividing and conquering the task in a clean and skillful way, and commenting on what you do are all illustrated nicely by Gödel's proof.
- Reading and understanding Gödel's proof yields many benefits. There are pearls to be found in its contents, and skills to be practiced that go beyond what one might think at first glance.

I am well aware that I did not give many arguments to support these opinions. That would be the stuff for a paper by itself, and the reader is encouraged to think about them. But above all, enjoy the paper "On formally undecidable propositions of Principia Mathematica and related systems I" itself, which after all makes up the main part of this document!

Literatur

- [1] Kurt Gödel. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. Dover, 1962.
- [2] Kurt Gödel. Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I. In Solomon Feferman, editor, *Kurt Gödel: Collected Works*, volume 1, pages 144–195. Oxford University Press, 1986. German text, parallel English translation.

B Index

- α , 8
- β , 8
- γ , 8
- ω -consistent, 14
- n -ary relation sign, 4
- argmin, 7
- subst, 4
- proof figure, 13

- axiom, 5

- basic sign, 4

- class-sign, 3
- comprehension axiom, 5
- consistent, 14

- decision-definite, 17
- degree, 7
- disjunction, 4

- elementary formula, 4

- formula, 4

- generalization, 4

- immediate consequence, 6

- negation, 4
- number-sign, 4

- P , 4
- Peano axioms, 5
- primitive recursion, 7
- primitive recursive, 7
- PM , 2
- proof, 2
- proposition-formula, 4
- proposition axioms, 5
- provable, 6

- quantor axioms, 5

- reducibility axiom, 5

- set axiom, 6
- sign of type n , 4
- sign of type one, 4

- type-lift, 5

- variable of type one, 4
- variable of type n , 4
- variable of type two, 4